
DevExpress Reports (WinForms, WPF, ASP.NET, MVC, .NET Core) v20.1 JSON Data Source Wizard – Parameter Expressions Video Script

Video URL: https://youtu.be/7fKqpJIX_dY

Product URL: <https://www.devexpress.com/subscriptions/reporting/>

Free Trial URL: <https://www.devexpress.com/Products/Try/>

DevExpress Reports – our royalty-free .NET reporting suite – ships with full support for all major .NET platforms including WinForms, WPF, ASP.NET, MVC, and .NET Core.

If you are new to DevExpress Reports or are considering our platform for an upcoming .NET project, be sure to point your browser to [devexpress.com/try](https://www.devexpress.com/try) to download your free 30-day trial today.

If you've used DevExpress Reports in the past, you already know that our reporting platform can retrieve data from a variety of data sources. In this particular video, I'll discuss support for JSON and demonstrate JSON Data Source Wizard-related enhancements we've introduced in our most recent release cycle – v20.1.

With that – let's get started.

As I mentioned a moment ago, DevExpress Reports allows you to connect to multiple data sources – be it a standard database, Entity Framework, Excel, or XPO.

I'll go ahead and select JSON from our Report Wizard data source type list to demonstrate. As you can see, our new JSON Report Wizard lets you specify expressions for a web service endpoint Path parameter, a Query parameter, and Header parameters. You can even bind these to report parameters as needed.

Let's now dig into implementation details. I'm going to bind my sample report to an OData Test Service. The link to this service is listed in the description.

([https://services.odata.org/V3/Northwind/Northwind.svc/Categories?\\$format=json&\\$filter=CategoryID%20eq%201](https://services.odata.org/V3/Northwind/Northwind.svc/Categories?$format=json&$filter=CategoryID%20eq%201))

Note that this particular JSON source supports data filtering via query parameters.

Ok – I'm now going to select the web service endpoint as a JSON source and specify the URI (<https://services.odata.org/V3/Northwind/Northwind.svc/>)

After entering the URI, I'll need to configure endpoint data. I'll create a path parameter with a static value pointing to the Categories collection. In this instance, I'll enter Categories for PathParameter1.

Before I get to Query, let's take a quick look at the bottom this wizard page. As you can see, our wizard displays the resulting JSON URI for you. It now includes both the URI and the Path parameter.

OK, let's keep going – time to add my Query parameters.

Note that this particular JSON source must include a static \$format parameter. Be certain to use query parameters names that are supported by your data server.

So, my first Query parameter is \$format and its value is json. My second parameter – let me quickly check it – ok, my second parameter is \$filter. Now for this parameter, I want to enter an Expression. I'll go ahead and mark the Expression checkbox and then I'll call the DevExpress Expression Editor. Here I can type: CategoryID eq 1.

Cool – I can leave this Expression unchanged, but I don't want to use a static value for this report. I want to use a dynamic value. To change this Expression so that it uses a user-provided dynamic value, I'll go ahead and return to the Query parameter list, and select New Report Parameter from the Value drop down list.

I'll go ahead and configure this new Report Parameter. I'll keep the default name, enter a description, specify data type and then I'll specify a default value just in case my users fail to enter a value. This will guarantee that data always loads from the service.

Ok – Now I'll open the Expression Editor once more and enter: CategoryID equals my new parameter. Great...almost done.

Time to close the Expression Editor and return to the Wizard page to check our URI. I'm happy with that, but before I click Next to proceed, a final note on parameters.

Path parameters and query parameters are included in endpoint requests in the same order as they are listed in our JSON Report Wizard. Move a parameter up or down in the list to change its position in endpoint requests.

Ok, onto the next step.

For this particular example, I don't need HTTP Authentication nor do I need to specify Header parameters. I'll go ahead and click Next once more and then Finish.

Ok – we're back at the DevExpress Report Designer – As you can see from my Field List, I have a simple data source I can use.

Time to specify the data member and simply drop fields onto the sample report's design surface to construct my simple JSON-based report.

Remember, since I entered a default parameter value, I can disable Request Parameters from my report. Let me go ahead and do that so I can preview this report.

Here it is – my simple report filtered by the default parameter value. I can use the DevExpress Parameters Panel to update parameter values as needed. When I make a change to a parameter value in this panel, the report goes straight to the data service endpoint to filter data and return the appropriate result set.

I hope you've enjoyed this short tutorial on DevExpress Reports and its new Localization Editor. If you liked this video, please give us a thumbs up. If you have questions, please comment below. And please remember to subscribe to this channel for more great DevExpress training videos.

Copyright © 2020 Developer Express Inc.

All trademarks or registered trademarks are property of their respective owners.